**GNU linker script，ld script，GNU链接脚本**

2016年03月20日 17:13:48 [itxiebo](https://me.csdn.net/itxiebo) 阅读数：4733

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/itxiebo/article/details/50937412

**[随笔]GNU linker script，ld script，GNU链接脚本**

注，此文部分内容摘抄自网络，博主(itxiebo)主要的贡献是整理编辑等。

**一、什么是GNU linker script？作用是什么？**

.lds文件，即linker script，决定了可执行映像（image）的链接方式，以及各个段的装载地址（装载域）和执行地址（运行域）。

1. 装载地址：运行之前各段的地址；
2. 运行地址：运行时各段的地址。

**二、对GNU linker script的简单认识**

首先，完整版的官网Gnulinkerscript下载地址如下，此文件可作为参考文献使用。   
<http://pan.baidu.com/s/1hriLMb6>

GNU官方网站上对.lds文件形式的基本描述：

SECTIONS

{

...

secname start BLOCK(align)(NOLOAD) : AT(ldadr)

{ contents } >region :phdr =fill

...

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7

两个必须的（secname, contents），其他optional

1. secname：段名，用以命名此段。
2. contents：决定哪些内容放在本段，可以是整个目标文件(.o)，也可以是目标文件中的某段（代码段、数据段等）。
3. start：是段的重定位地址，即本段运行的地址。如果代码中有位置无关指令，程序运行时这个段必须放在这个地址上。start可以用任意一种描述地址的符号来描述。
4. BLOCK(align) 指定块对齐。比如，前一个段从0x30000000到0x300003F1，此处标记ALIGN(4)，表示此处最小占用4Bytes，即使下一个段是紧挨这个段，那么下一个段的起始地址（也就是运行地址）为0x300003F4。
5. NOLOAD：告诉加载器程序运行时不加载该段到内存。
6. AT(ldadr)：定义本段存储（加载）的地址，如果不使用这个选项，则加载地址等于运行地址，通过这个选项可以控制各段分别保存于输出文件中不同的位置。

**三、.lds相关的简单实例**

例：   
/\* nand.lds \*/

SECTIONS

{

first 0x00000000 : { head.o init.o }

second 0x30000000 : AT(4096) { main.o }

}

* 1
* 2
* 3
* 4
* 5

以上，

head.o放在0x00000000地址开始处，init.o放在head.o后面，他们的运行地址也是0x00000000，即存储和运行地址相同（没有AT指定）；

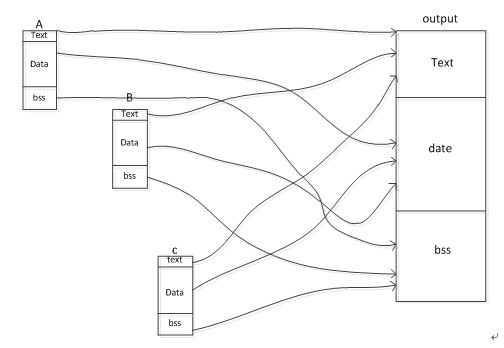
main.o放在4096（0x1000，是AT指定的，存储地址）开始处，但它的运行地址在0x30000000，运行之前需要从0x1000（加载地址处）复制到0x30000000（运行地址处），此过程也就需要读取flash，把程序拷贝到相应位置才能运行。这就是存储地址和运行地址的不同，称为加载时域和运行时域，可以在.lds连接脚本文件中分别指定。

另外，编写好的.lds文件，在用arm-linux-ld链接命令时：

1. 带-Tfilename来调用执行，如arm-linux-ld -Tnand.lds x.o y.o -o xy.o
2. 也用-Ttext参数直接指定链接地址，如arm-linux-ld -Ttext 0x30000000 x.o y.o -o xy.o

**四、常用section的定义**

1. text section：存放程序代码
2. data section：存放数据
3. bss section：存放未初始化的数据，在镜像文件中，是不为 bss 段分配空间的，所以如果你开一个很大的全局的未初始化的数组，镜像文件的大小不会相应的变大。而只是在加载器将镜像加载进内存时，才会为 bss 段分配空间

链接器：把各个目标文件的各种段进行重新组合。   


最后，两个命令，先保存起来，后面有机会再实践一下。

1. arm-linux-objdump -h vmlinux > vmlinux.txt   
   –输出linux内核段信息到vmlinux.txt文本中。
2. arm-linux-objdump -f vmlinux > sec\_symbol.txt   
   –输出内核段信息和符号到sec\_symbol.txt文本中。

本文回溯

1. 什么是GNU linker script？作用是什么？
2. 对GNU linker script的简单认识，核心SECTIONS的结构分析。
3. 对链接器基本原理的理解。

完